

## **CLAIM AMENDMENTS**

### **Claim Amendment Summary**

#### **Claims pending**

- Before this Amendment: Claims 1-5 and 18-24
- After this Amendment: Claims 1-5 and 18-24

**Non-Elected, Canceled, or Withdrawn claims:** None

**Amended claims:** None

**New claims:** None

---

### **Claims:**

1. (Previously Presented) A method for authorizing an access to a table of address correspondence between a multitask CPU and at least one memory containing several programs, comprising:

calculating, on each task change between a first program module switching from foreground to background and a second program module switching from background to foreground, a signature of at least part of the second program module instruction lines, and

checking the conformity of this signature with a signature recorded upon previous foreground execution in the of the second program module, each signature being associated with a program identifier.

2. (Original) The method of claim 1, wherein said signature is calculated by the implementation of a Hash function.

3. (Original) The method of claim 1, wherein said memory is a RAM in which are loaded program lines from a mass storage.

4. (Previously Presented) A processor of multitask execution of several programs, each of the several programs being different from each other, exploiting a table of correspondence between virtual addresses of the lines of the different programs and physical addresses of these lines in at least one memory, each correspondence being associated with an identifier of the involved program when executed, comprising means for calculating a current signature based on at least part of the program lines in said memory, and means for comparing this signature with the identifier of the program stored in the correspondence table.

5. (Original) The processor of claim 4, wherein the identity of the signature and of the program identifier allows the CPU to execute the instruction of the involved program.

6 -17. (Canceled)

18. (Previously Presented) A method comprising:  
executing, at a CPU, a plurality of programs simultaneously, each program having a unique signature calculated when first executed, wherein each program includes currently-executing tasks that change between a foreground and a background at the CPU;

calculating, on each task change, a new signature of at least part of program instruction lines for the program associated with the task; and

checking the conformity of the new signature with the unique signature.

19. (Previously Presented) The method of claim 18, wherein each signature is calculated by the implementation of a Hash function.

20. (Previously Presented) The method of claim 18, further comprising suspending execution of a program if the new signature and the unique signature do not conform.

21. (Previously Presented) The method of claim 18, further comprising:  
storing the new signature in a memory; and  
checking the conformity of a next new signature with the stored new signature at the next task change associated with the program.

22. (Previously Presented) The method of claim 18, further comprising establishing the unique signature when the associated program is first executed.

23. (Previously Presented) The method of claim 18, further comprising:  
storing the unique signature in a memory; and  
checking the conformity of a next new signature with the stored unique signature at the next task change associated with the program.

24. (Previously Presented) The method of claim 18, further comprising:  
storing the new signature in a memory in response to a first task change;  
and  
checking the conformity of the new signature with the unique signature in  
response to a second task change.